Decision Tree as Knowledge Management Tool in Image Classification

Kusrini[a], Agus Harjoko[b]

[a]*STMIK AMIKOM Yogyakarta*
Jl. Ringroad Utara Condong Catur Sleman Yogyakarta Indonesia
Tel : 62-274-884201, Fax : 62-274-884208
E-mail : kusrini@amikom.ac.id

[b]*Computer Science Program Faculty of Mathamtics and Natural Science*
Gadjah Mada University, Yogyakarta, Indonesia
Tel : 628164267256, Fax : 62-274-545185
E-mail : aharjoko@ugm.ac.id

ABSTRACT

Expert System has been grown so fast as a science that study how to make computer capable of solving problems that typically can only be solved by expert. It has been realized that the biggest challenge of developing expert system is the process include expert's knowledge into the system. This research tries to model expert's knowledge management using case based reasoning method. The knowledge itself is not inputted directly by the expert, but rather the system will learn the knowledge from what the expert did to the previous cases. This research takes image classification as the problem to be solved. As the knowledge development technique, we build decision tree by using C4.5 algorithm. Variables used for building the decision tree are the image visual features.

Keywords

Expert System, Case Based Reasoning, Knowledge Management, Image Classification

1.0 INTRODUCTION

Expert System has been grown so fast as a science that study how to make computer capable of solving problem that typically can only be solved by expert. The biggest challenge of developing expert system is the process include the expert's knowledge into the system.

This research tries to model expert knowledge management using case based reasoning method. The knowledge itself is not inserted directly by an expert. The system will learn the knowledge fom what the expert did in the previous cases. As the case study, we take image classification problem.

An image can be recognized visually from its image visual feature. The most frequent features to be used to describe an image are *color* (histogram, mo ment, linguistic tag, etc), *shape* and *texture* (Acharya & Ray, 2005)

Earlier research had been performed to build visual image retrieval application that used single feature. Among features of color histogram, moment and linguistic color tag; the histogram color feature showed the best performance. However, using single feature was not satisfaction enough to represent an image.

In this research, we try to represent image by using a number of features simultaneously. These features are then used as variable to classify the image. We use inductive method by building decision tree. In our previous research showed that decision tree using C4.5 algorithm gave better performance for text data classification than nearest neighbor method.

C4.5 algorithm has been implemented to evaluate the Cancellation Possibility of new student applicants at STMIK AMIKOM Yogyakarta (Kusrini & Hartati, 2007).

The decision tree is built by using result of classification of previous cases classification as the basis to generate the new one.

Finally, the goal of this research is to develop an application that can be used to classify image based on previous cases.

2.0 THEORITICAL BACKGROUND

2.1 Case Based Reasoning

Case-based reasoning (CBR) is a problem solving technique based on previous experience knowledge (Armengol, Onta & Plaza, ---).

The problem-solving cycle in a CBR system consists of the following 4 processes (see Fig. 1):

1. Retrieving similar previously experienced cases (e.g., problem–solution–outcome triples) which problem is judged to be similar
2. Reusing the cases by copying or integrating the solutions from the cases retrieved
3. Revising or adapting the solution(s) retrieved in an attempt to solve the new problem
4. Retaining the new solution once it has been confirmed or validated

Assumptions that used in case based reasoning are:
1. Similar problem will have similar solution
2. World is a constant place. Whatever things that have a true value today will always have a true value later
3. Situation is repeated

Case based reasoning is very good to be implemented in classification like diagnosis (medical) and prediction, but also hard to implement for synthetic working like designing, planning and scheduling (Berry & Linoff, 2004)

2.2 Decision Tree

A decision tree model consists of a set of rules for dividing a large heterogeneous population into smaller, more homogeneous groups with respect to a particular target variable (Larose, 2005).

The target variable is usually categorical and the decision tree model is used either to calculate the probability that a given record belongs to each of the categories, or to classify the record by assigning it to the most likely class. Decision trees can also be used to estimate the value of a continuous variable, although there are other techniques more suitable to that task (Larose, 2005).

Kusrini has been implement C4.5 algorithm to build decision tree to analyze cancellation of student candidate registration in STMIK AMIKOM Yogyakarta (Kusrini, 2007).

The C4.5 *algorithm* is Quinlan's extension of his own ID3 algorithm for generating decision trees (Berry & Linoff, 2004). Just as with CART, the C4.5 algorithm recursively visits each decision node, selecting the optimal split, until no further splits are possible. However, there are interesting differences between CART and C4.5 (Larose, 2005):
- Unlike CART, the C4.5 algorithm is not restricted to binary splits. Whereas CART always produces a binary tree, C4.5 produces a tree of more variable shape.
- For categorical attributes, C4.5 by default produces a separate branch for each value of the categorical attribute. This may result in more "bushiness" than desired, since some values may have low frequency or may naturally be associated with other values.
- The C4.5 method for measuring node homogeneity is quite different from the CART method and is examined in detail below.

2.3 C4.5 Algorithm

In general, steps in C4.5 algorithm to build decision tree are (Craw, 2005):
- Choose attribute for root node
- Create branch for each value of that attribute
- Split cases according to branches
- Repeat process for each branch until all cases in the branch have the same class

Choosing which attribute to be a root is based on highest gain of each attribute. To count the gain, we use formula 1, below (Craw, 2005):

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} * Entropy(Si) \quad (1)$$

with {S1, ..., Si, …, Sn} *is* partitions of S according to values of attribute A, n is number of attributes A, |Si| is number of cases in the partition Si and |S| is total number of cases in S

While entropy is gotten by formula 2 below (Craw, 2005):

$$Entropy(S) = \sum_{i=1}^{n} - pi * \log_2 pi \quad (2)$$

With S is case set, n is number of cases in the partition S and p_i is proportion of S_i to S

2.4 Digital Image

Digital image is define as function f(x,y), where x and y denote spatial coordinates and the value f at any point (x,y) is proportional to the brightness (or gray level) of the image at the point (Gonzalez & Paul, 1987).

2.5 Image Feature Extraction

An image can be recognized visually from their features. Some features that can be extracted from an image are color, shape and texture.

Color has been successfully applied to retrieve images, because it has very strong correlations with the underlying objects in an image. Moreover, color feature is robust to background complications, scaling, orientation, perspective, and size of an image (Acharya & Ray, 2005).

Some features that can be extracted from image color are color histogram, color moment and linguistic color tag (Acharya & Ray, 2005).

Color histogram is a color feature that most widely used. Color histogram is effective to characterize global distribution of image color.

To define color histograms, the color space is quantized into a finite number of discrete levels. Each of these levels becomes a bin in the histogram. The color histogram is then computed by counting the number of pixels in each of these discrete levels.

Color Moment is a compact representation of the color feature to characterize a color image. It has been shown

that most of the color distribution information is captured by the three low-order moments. The first-order moment *(μ)* captures the mean color, the second-order moment *(s)* captures the standard deviation, and the third-order moment captures the skewness *(?)* of color. These three low-order moments (*μc, sc, ?c*) are extracted for each of the three color planes, using the following mathematical formulation. Formula of first-order moment is shown in formula 3, formula of second-order moment is shown in formula 4 and Formula of third-order moment is shown in formula 5 (Acharya & Ray, 2005).

$$m_c = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} p_{ij}^{c} \quad (3)$$

$$s_c = \left[\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(p_{ij}^{c} - m_c)^2\right]^{\frac{1}{2}} \quad (4)$$

$$q_c = \left[\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(p_{ij}^{c} - m_c)^3\right]^{\frac{1}{3}} \quad (5)$$

Texture is the characteristic belong to a region that is big enough for repetition of the characteristic itself. Texture also can be defined as a regularity of specific patterns that is formed from pixels construction in digital image.

A surface is considered to have texture information only if the region is enlarged without change the scale, then the surface characteristic of enlargement result, have similarity with the original surface.

Texture information can be used to distinguish the object surface characteristic in image that related with coarse and smooth without affected by the color.

Texture can exist if:

1. Existence of primitive pattern consists of one or more pixels, these primitive patterns can be a dot, straight line, curve line, region, etc. which is the basic element of a shape.
2. Those primitive patterns appear repeatedly with specific distant interval and direction so that it can be predicted or found the repetition characteristic.

Texture information can not be obtained by using one dimensional histogram. To get the intensity values location relationship or dependency that is very important in texture perception, we need two-dimensional relational matrix that is known as co-occurrence intensity matrix.

Co-occurrence intensity matrix is a matrix that representing the occurrence frequency of two pixel pair with specific intensity in specific distance and direction in an image. Co-occurrence intensity matrix $p(i_1,i_2)$ is defined with two simple process below:

a. Determine the distant between two dots in direction of vertical and horizontal (vector d=(dx,dy)). The values of dx and dy are declared in pixel as the smallest unit in digital image.
b. Calculate pixels pairs that have intensity values i_1 and i_2 and have d pixel distance in image. The calculation result of every intensity value pair is then placed on the matrix in appropriate coordinate, with i_1 as abscissa value and i_2 as ordinate value.

Texture features that can be extracted from the image are like entropy, energy, contrast, and homogeneity.

Entropy is a feature for measuring disorderly of intensity distribution. The formula that can be used to calculate entropy is shown in formula 9.

$$Entropy = -\sum_{i_1}\sum_{i_2} p(i_1,i_2)\log\, p(i_1,i_2) \quad .....(9)$$

Energy is a feature for measuring intensity pair concentration on co-occurrence matrix. The formula that is used to calculate energy is shown in Formula 10. The energy value will be increased if the pixel pair that satisfy the requirement of co-occurrence intensity matrix is concentrated to some coordinates and be decreased if the location is dispersed.

$$Energy = \sum_{i_1}\sum_{i_2} p^2(i_1,i_2) \quad .....(10)$$

Contrast is a feature for measuring intensity strength difference in an image. The contrast value will be increased if image intensity variation is high and it will be decreased if the variation is low. The formula to measure the image contrast is shown in Formula 11.

$$Contrast = \sum_{i_1}\sum_{i_2}(i_1 - i_2)^2\, p(i_1,i_2) \quad .....(11)$$

Homogeneity is used to measure homogeneity of image intensity variation. The value of homogeneity will be increased if the variation intensity in image is decreased. Homogeneity is calculated with Formula 12

$$Homogeneity = \sum_{i_1}\sum_{i_2}\frac{p(i_1,i_2)}{1+|i_1 - i_2|} \quad .....(12)$$

p Notation in Formula 10, 11 and 12 is denoted the probability in range of 0 to 1, that is the element value in co-occurrence matrix. Meanwhile i_1 and i_2 are denoted nearby intensity pair that in co-occurrence matrix, they will be row and column number respectively.

3.0 SYSTEM DESIGN

Features that are used as classification variable in this research are first order color moment, second order color

moment, third order color moment, entropy, energy, contras and homogeneity. We choose these features because they have single value for every image. It is different with color histogram that has array value. Color histogram is difficult to implement in our research.

This application uses 2 kinds of tables; they are initial table and running table. Initial table is tables that create when the application is developed, while running table is tables that generate by system when application is running.

The initial tables in the application are:

1. Table *Variable_List*. This table has 3 attributes; they are *variable name*, *is_result* and *is_active*. Table *Variable_List* is used to store list of variable that used to make decision tree. *Is_Result* is told about whether the variable is a result variable or not, while *is_active* is flagged whether the variable is used or not.

2. Table *Variable_Value*.This table is used to store list of variable value for every variable in *variable_list* table. The function of this table is discretize the continues value of variable value. It has four attributes, they are: *Variable_name*, *Variable_Value*, *Lower_limit* and *Upper_Limit*

3. Table *Case*. This table has n attributes, they are *variable_name[1]*, *variable_name[2]*, …, *variable_name[n]*. n is representing count of record in the *variable_list* table that has value of *is_active* is True, while *variable_name[1]*, *variable_name[2]*, …, *variable_name[n]* is value of *variable_name* that has value of *is_active* is True. For example, value of variable list table is shown in Table 1.

Table 1. Table Variable_List

Variable_name	**Is_result**	**Is_active**
Entropy	False	True
Energy	False	True
Contras	False	False
Class	True	True

Based on value of table *variable_list* cases table will have 3 attribute, they are *Entropy*, *Energy* and *Class*.

4. Table *Tree*. This table has 5 attributes; they are *id_node*, *node*, *prime*, *value* and *is_variable*. This table is used for store result of decision tree making process.

There are two kinds of running table in our application, they are:

1. Table *Work[0] .. Work[n-1]*. Attributes of each work table are *Variable_name* and *gain*. Each work table is created for every calculation cycle to store variable and its gain value.

2. Table *Sub_work[0]…Sub_work[n-1]*. Attributes of each sub_work table are *Variable_Name*, *Value*, *Result[1]*, …, *Reslt[n]*, *count*, *and entropy*. Each *sub_work* table is created for every calculation cycle to store variable and value. *Result* attribute are value in *result_variable* of *case* table. Each *result*, *count* and *entropy* attribute is calculate for every combination of *variable_name* and *value* attribute

4.0 RESULT

The result of this research is an application to classify image based on result of classification of previous case classification. Menu Structure of this application is shown in Figure 1.

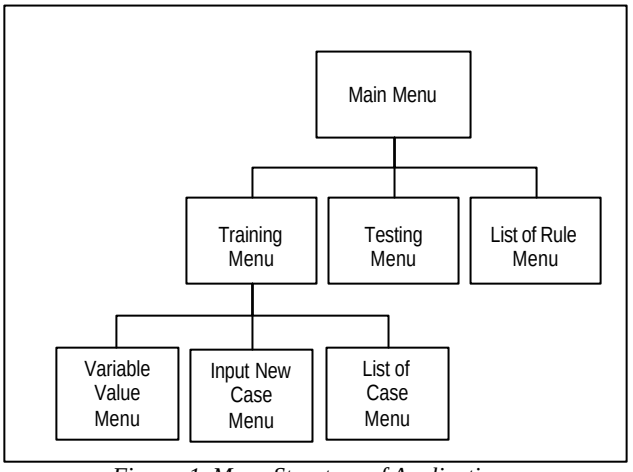

Figure 1. Menu Structure of Application

The interface for training is shown in Figure 2.

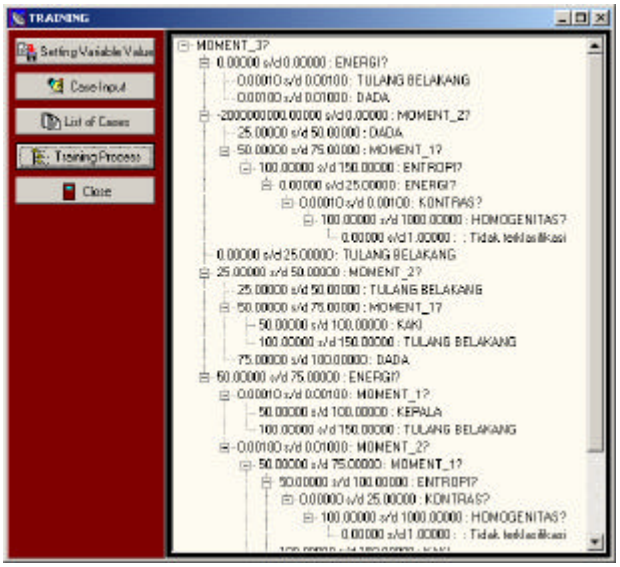

Figure 2. Training Interface

Result that is shown in Figure 3 was taken after *Training Process* Button was clicked. This tree then saved into table *tree*.

The values of tree table are then presented as a list of rule as shown in Figure 3.

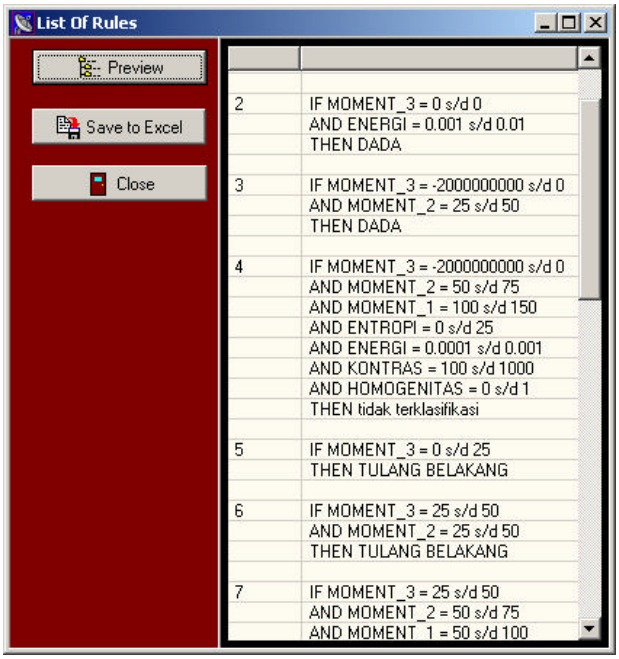

Figure 3. List of Rule

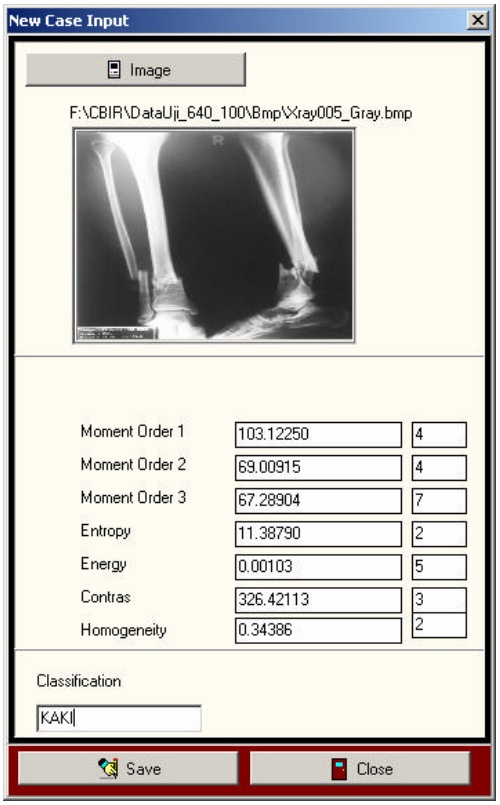

Figure 4. Input New Case Interface.

Training process can be started after variable value has been set, and the case has been inputted. The interface of new case input is shown in Figure 4.

To input new case, user has to choose image before the system calculate image visual features. After that, user will need to enter classification of this image.

To classify a new image, the user only have to enter image file location then system will calculate image visual features and the classification based on decision tree that has been build in training process.

This application has been tested for syntax error, runtime error and logical error.

For accuracy test, we divide the case into old case that has been inputted into system and new case that is not present in case database.

The first case showed that 100% of case is classified into truth classification. Meanwhile the second one showed some possibilities, they are:

1. The case is classified into truth classification.
2. The case is classified into wrong classification. The accuracy level is depended to chosen feature as classification variable and the definition/setting of variable value
3. The case is not classified, because after all variables are traveled until leaf node of the tree, class is not homogeneous
4. The case is not classified, because variable value of the new case has not yet been defined in case database.

In this research, we do not handle automatic case adaptation when there is a change in variable value. As the consequences, new case input process can only be made after all variable values are set and training process can only be made after all cases have been inputted to database. Any single change to the case's variable value will make the case have to be re-inputted and re-trained.

This is considered to be a weak point of this application and will be handled in next research.

5.0 SUMMARY

Decision tree can be used as a knowledge management tool for classification of digital image.

The application for image classification has been developed with facilities of training and testing. Training facility is used to input previous case into case database and build decision tree when the testing facility is used to classify new image into a class based on decision tree built before (in training process). Knowledge of image classification can be accessed from list of rule facility.

REFERENCES

Armengol, E., Onta, S., dan Plaza, E. (---). *Explaining similarity in CBR Eva Armengol*, Artificial Intelligence Research Institute (IIIA-CSIC). Campus UAB, 08193 Bellaterra, Catalonia

Berry, Michael J.A., Linoff, Gordon S. (2004). *Data Mining Techniques For Marketing, Sales, and Customer RelationshipManagement. Second Edition.* Wiley Publishing, Inc., Indianapolis, Indiana

Craw, S. (2005). *Case Based Reasoning : Lecture 3: CBR Case-Base Indexing*. www.comp.rgu.ac.uk/staff/smc/teaching/cm3016/Lecture-3-cbr-indexing.ppt

Gonzalez, Rafael C & Paul A (1987), *Digital Image Processing.* Addison-Wesley Publishing Company, Inc. Canada.

Larose, Daniel T. (2005). *Discovering Knowledge in Data: an Introduction to Data Mining.* John Wiley and Sons, USA

Kusrini, (2007), *Penggunaan Pohon Keputusan untuk Menganalisis Kemungkinan Pengunduran Diri Calon Mahasiswa Baru di STMIK AMIKOM Yogyakarta* (Prosiding Seminar Nasional Teknologi 2007, ISSN 1978-9777)

Kusrini, Hartati, S. (2007). *Implementation of C4.5 Algorithm to evaluate the Cancellation Possibility of New Student Applicants.* Procedings of The International Conference on Electrical Engineering and Informatics.

Pall, Sankar K., Shiu, Simon C.K.,. (2004). *Foundation of Soft Case Based Reasoning.* John Wiley and Sons, USA